Movie Recommendation System – Internship Report

Intern: Sneha Kumari

---

1. Introduction

With the growth of streaming platforms, users face **content overload**, making it difficult to choose what to watch. This project develops a **Movie Recommendation System** using **machine learning and content-based filtering** to suggest movies based on their content, including overview, genres, cast, keywords, and crew. Unlike popularity-based approaches, the system focuses on **meaningful, context-aware recommendations**.

---

2. Data Collection & Preprocessing

Datasets Used:

- `tmdb_5000_movies.csv` and `tmdb_5000_credits.csv` (Kaggle)

APIs: OMDB (for movie posters)

Preprocessing Steps:

1. Removed null and duplicate entries.
2. Extracted relevant features: overview, genres, cast, keywords, crew.
3. Combined features into a single **tags** column.
4. Converted text to numeric vectors using **CountVectorizer**.

---

3. Methodology

The system uses **Cosine Similarity** to measure closeness between movies in the vector space.

Workflow:

1. User inputs a movie title.
2. Tags are vectorized.
3. Cosine similarity is computed with all movies.
4. Top 5–10 similar movies are recommended with posters via **Streamlit**.

This approach captures **thematic, stylistic, and narrative similarity**, providing contextually relevant suggestions.

---

4. Results & Evaluation

Example:

- **Input:** Avatar (2009)
- **Recommendations:** Aliens vs Predator: Requiem, Falcon Rising, Aliens, Independence Day, Titan A.E.

Evaluation Metrics:

Metric	Score
Precision@10	0.50
Recall	1.00
F1-Score	0.67
Cosine Similarity	0.6–1.0

Heatmaps show **strong self-similarity** and clusters of thematically similar movies.

---

5. Technology Stack

- **Language:** Python
- **Libraries:** Pandas, NumPy, Scikit-learn, NLTK

- **ML Techniques:** CountVectorizer, Cosine Similarity
- **Deployment:** Streamlit, Pickle
- **APIs/Datasets:** OMDB, Kaggle `tmdb_5000_movies` & `tmdb_5000_credits`

---

6. Discussion & Future Work

- The system effectively recommends contextually similar movies without using user data.
- Future enhancements:
  - Hybrid filtering (content + user behavior)
  - BERT embeddings for semantic understanding
  - Poster/image similarity with CNNs
  - Cloud deployment for scalability

The model is lightweight, privacy-respecting, and **sustainable for real-time recommendation**.

---

7. Conclusion

The Movie Recommendation System helps users **discover movies aligned with their preferences** efficiently. By leveraging **machine learning and NLP**, it provides accurate, context-aware suggestions while being scalable and privacy-friendly.

"Helping users find the right movie at the right time."